

GAmEraHOM

Un generador de mutantes de orden superior para composiciones WS-BPEL

Emma Blanco Muñoz

Ingeniería en Informática

23 de enero del 2012



Este documento está bajo la licencia Creative Commons Atribución 3.0 Unported.

Índice

Sobre mi

Motivación y objetivo

Planificación

Antecedentes

Arquitectura de GAmEraHOM

Calidad

Demostración práctica

Publicaciones

Conclusiones y trabajo futuro

¿Quién soy? ¿de dónde vengo? ¿a dónde voy?

Historial

- 2005/09 Ingeniería Técnica en Informática de Gestión
- 2009/11 Ingeniería Informática
 - **Beca de colaboración** del Ministerio de Educación
 - Lenguajes y Sistemas Informáticos
Grupo **UCASE** de Ingeniería del Software
- 2011/- Itinerario en Modelado, Simulación y Pruebas de Procesos y Tratamiento de Señales y Datos



¿Por qué este proyecto?

WS-BPEL

- Amplio uso de composiciones de servicios WS-BPEL.
- WS-BPEL es un reto para las técnicas de caja blanca porque incluyen instrucciones específicas para SW no comunes en otros lenguajes.

Sistemas de generación de mutantes

Primer orden SQLMutation, Mothra,
MuJava, Proteum,
GAmera...

Orden superior MILU.



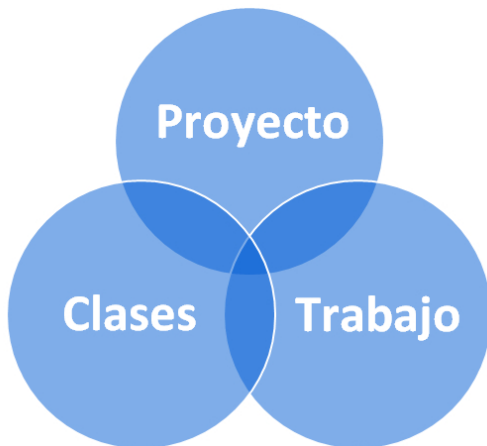
Meta a conseguir

Objetivo

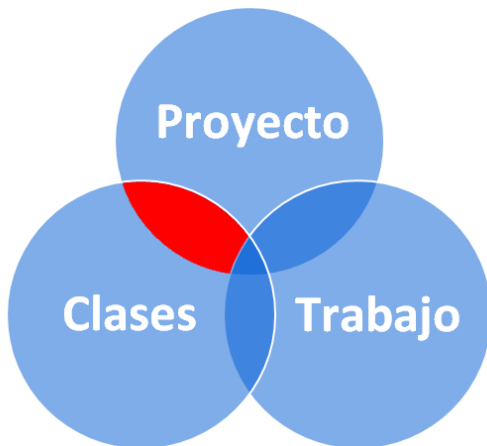
Primer generador automático de mutantes de orden superior para composiciones WS-BPEL.

algoritmos aptitud campo codificacion composiciones cruce datos estructura
gamera generacion generador geneticos herramientas
mutacion individuos mutantes
operador orden operadores primer
secuencia seleccion superior tipos ws-bpel

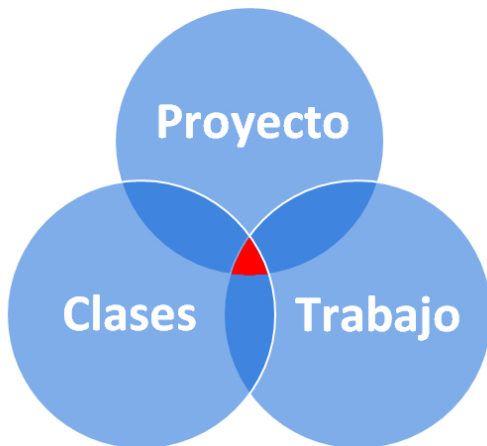
Planificación



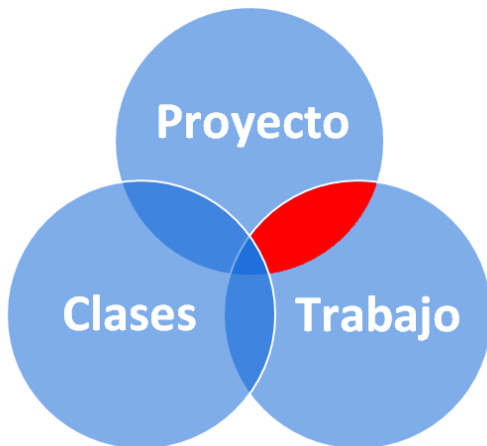
Planificación



Planificación



Planificación



Poniéndonos al día

Prueba de mutaciones

Mutante Programa a partir del original con modificaciones sintácticas.

Programa original
`amount < 10000`

Mutante
`amount > 100000`

Mutante vivo Si ningún caso de prueba puede diferenciar el mutante del programa original (0).

Mutante muerto Si un caso de prueba puede distinguirlos (1).

Mutante erróneo Si se produce un error en el despliegue (2).

Poniéndonos al día

Algoritmos genéticos

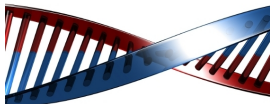
Individuo Posible solución al problema.

Población Conjunto de individuos.

Aptitud Calidad de la solución del problema a resolver.

Generación Cada iteración del AG en búsqueda de la solución óptima.

Op. genéticos Selección y reproducción (mutación y cruce).



Poniéndonos al día

Algoritmos genéticos

Individuo Posible solución al problema.

Población Conjunto de individuos.

Aptitud Calidad de la solución del problema a resolver.

Generación Cada iteración del AG en búsqueda de la solución óptima.

Op. genéticos Selección y reproducción (mutación y cruce).

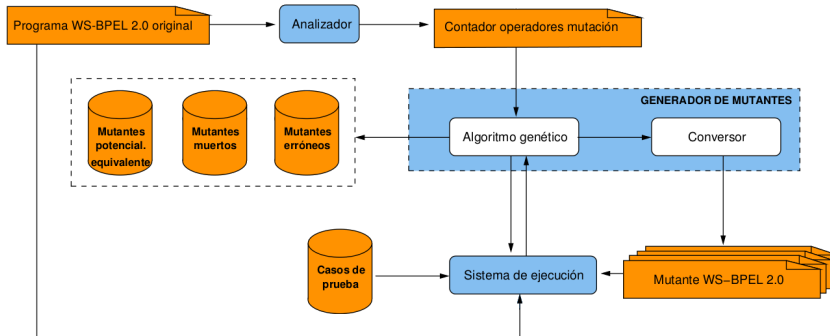
Mutación evolutiva

- Genera sólo un **subconjunto** de todos los posibles mutantes sin pérdida significativa de información.
- Basada en algoritmos genéticos.

Poniéndonos al día

GAmEra

- Generador de mutantes de primer orden para WS-BPEL.
- Escrito en C++.

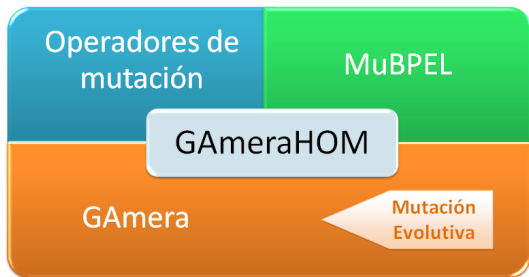


GAmera VS GAmeraHOM

Características	GAmera	GAmeraHOM
Lenguajes	C++	Java SE
Mantenimiento	Difícil	Fácil
Configuración	Dependiente	Independiente
Modularidad	Baja	Alta

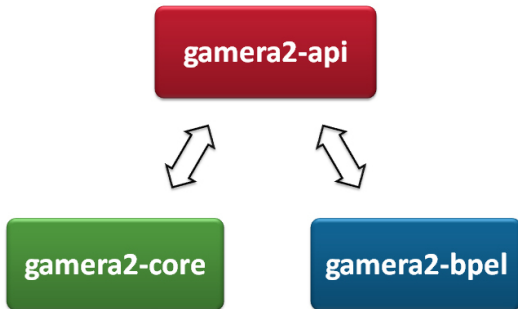
GAmera VS GAmeraHOM

Características	GAmera	GAmeraHOM
Lenguajes	C++	Java SE
Mantenimiento	Difícil	Fácil
Configuración	Dependiente	Independiente
Modularidad	Baja	Alta



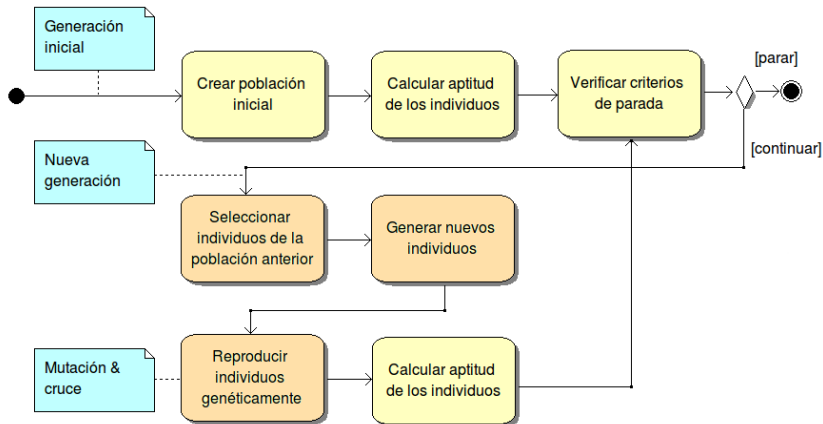
Arquitectura de GAmEraHOM

División de componentes



Arquitectura de GAmEraHOM

Flujo del algoritmo genético



Arquitectura de GAmEraHOM

Nueva estructura del individuo

Orden máximo Número máximo de mutaciones.

Orden Número de mutaciones reales del individuo.

Cada mutación estará compuesta por:

Operador Operador de mutación que se aplica.

Localidad Número de instrucción implicada en la mutación.

Atributo Información complementaria del operador.

Individuo \mathcal{A}

orden = 3

aptitud = a_1

7	1	5	7	9
5	4	8	4	3
2	3	1	1	6

op

loc

atr

Arquitectura de GAmeraHOM

Igualdad de individuos

- GAmeraHOM permite que una misma población tenga individuos de distintos órdenes.
- Ésto plantea la posibilidad de tener individuos genéticamente diferentes, pero que representan al mismo mutante.

Individuo \mathcal{A}

orden = 3

aptitud = a_1

7	1	5	7	9
5	4	8	4	3
2	3	1	1	6

op

loc

atr

Individuo \mathcal{B}

orden = 3

aptitud = a_2

7	1	5	5	5
5	4	8	3	6
2	3	1	1	9

op

loc

atr

Arquitectura de GAmeraHOM

Cálculo del fitness

$$Fitness(I) = M \cdot T - \sum_{j=1}^T \left(m_{lj} \cdot \sum_{i=1}^M m_{ij} \right) \quad (1)$$

Salón de la fama (HOF)

Registro de los mejores individuos encontrados por el AG.

Operadores genéticos

- Mutación** Nuevo individuo a partir de la modificación de un campo de un individuo.
- Cruce** Dos nuevos individuos por el intercambio de valores de dos individuos.

Arquitectura de GAmeraHOM

Mutación de orden

Individuo \mathcal{A}

orden = 3

aptitud = a_1

7	1	5	7	9	op
5	4	8	4	3	loc
2	3	1	1	6	atr

$$\text{Orden}(I) = o_{\text{actual}} + \text{rand}(-\text{rango}, \text{rango}) \cdot (1 - \text{prob}_{MO})$$

Arquitectura de GAmeraHOM

Mutación de individuo

Individuo \mathcal{A}

orden = 3
aptitud = a_1

7	1	5	7	9	op
5	4	8	4	3	loc
2	3	1	1	6	atr

Punto de mutación

$$\text{Valor}(l) = v_{\text{actual}} + \text{rand}(-\text{rango}, \text{rango}) \cdot (1 - \text{prob}_{MI})$$

Arquitectura de GAmEraHOM

Cruce de orden

Individuo \mathcal{A}

orden = 3

aptitud = a_1

7	1	5	7	9	op
5	4	8	4	3	loc
2	3	1	1	6	atr

Individuo \mathcal{C}

orden = 3

aptitud = a_2

7	4	5	7	3	op
5	4	1	4	3	loc
3	3	5	1	5	atr

Punto de cruce



7	1	5	7	3	op
5	4	1	4	3	loc
2	3	5	1	5	atr

7	4	5	7	9	op
5	4	8	4	3	loc
3	3	1	1	6	atr

Arquitectura de GAmEraHOM

Cruce de individuo

Individuo \mathcal{A}

orden = 3

aptitud = a_1

7	1	5	7	9	op
5	4	8	4	3	loc
2	3	1	1	6	atr

7	4	5	7	9	op
5	4	8	4	3	loc
2	3	1	1	6	atr

Individuo \mathcal{C}

orden = 3

aptitud = a_2

7	4	5	7	3	op
5	4	1	4	3	loc
3	3	5	1	5	atr



7	1	5	7	3	op
5	4	1	4	3	loc
3	3	5	1	5	atr

Punto de cruce

sonar , por un código de buena calidad

Lines of code

1,396

2,409 lines
578 statements
22 files

Classes

22

8 packages
96 methods
+37 accessors

Violations

1

Rules compliance

99.8%



Blocker

0



Critical

0



Major

1



Minor

0



Info

0



Comments

26.3%

499 lines
100.0% docu. API
0 undocu. API
0 commented LOCs

Duplications

0.0%

0 lines
0 blocks
0 files

Package tangle index

0.0%

sonar, por un código de buena calidad

es.uca.webservices.gamera.cli.CLIRunner

Coverage Dependencies Duplications LCOM4 Source Violations

1 violations Blocker: 0 Critical: 0 Major: 1 Minor: 0 Info: 0

☐ Full source | Time changes... | All violations

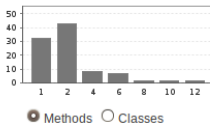
```
68      */
69      static int mainMethod(String[] args) {
70          if (args.length != 1) {
71              try {
72                  System.err.println(String.format(
73                      "Gamera version %s\nUsage: %s (.yaml file)",
74                      getProperty("gamera.version"),
75                      getProperty("launcher.name"));
76              }
              return 1;
          }
      }
```

System Println | 26 días

System.out.print is used

sonar, por un código de buena calidad

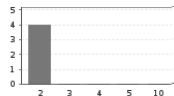
Complexity
2.9 /method
12.6 /class
12.6 /file
Total: 277



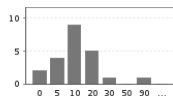
Code coverage
88.7% ▲
90.4% line coverage ▲
84.2% branch coverage ▲

Test success
100.0%
0 failures
0 errors
95 tests
7:17 min ▲

LCOM4
1.2 /class
18.2% files having LCOM4>1



Response for Class
19 /class



sonar, por un código de buena calidad

 es.uca.webservices.gamera.genetic.IndividualCrossoverOperatorTest

[Source](#)

Tests

100%

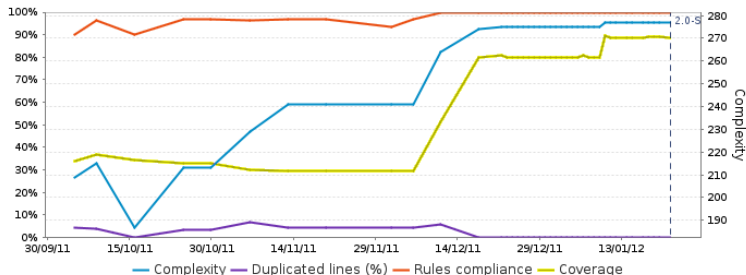
Tests: 6

Failures/Errors: 0/0

Duration: 10 ms

Duration	Unit test name
 1 ms	testCrossover
 9 ms	applyingValidIndividualCrossover
 0 ms	applyingWrongMutatedField
 0 ms	applyingWrongIndividualCrossover_CPisZero
 0 ms	applyingWrongIndividualCrossover_CPisGreaterThanMaxOrder
 0 ms	applyingWrongIndividualCrossover_CPisGreaterThanParentsMin

sonar, por un código de buena calidad



sonar, por un código de buena calidad

	Dependency	Suspect dependency (cycle)	- uses >	- uses >	
es.uca.webservices.gamera.cli	-				
es.uca.webservices.gamera.generate	-				
es.uca.webservices.gamera.genetic	-				
es.uca.webservices.gamera.select	1	-			
es.uca.webservices.gamera.term	-				
es.uca.webservices.gamera	1			-	
es.uca.webservices.gamera.conf	1			1	-
es.uca.webservices.gamera.log				1	-



Jenkins, para la integración

Jenkins

Jenkins - GAmEra



[Personas](#)



[Historial de construcción](#)



[Dependencia entre proyectos](#)



[Comprobar firma de ficheros](#)























Trabajos en cola

No hay trabajos en cola

Estado de los nodos

#	Principal
1	Inactivo
Ubuntu Server 64 bits	
1	Inactivo



All	BPELUnit	Esper	GAmEra	Takuan			
S	W	Name ↓	Último éxito	Último fallo	Última duración		
		bpel-packager	18 Hor (#106)	N/D	41 Seg		
		gamera2-api	18 Hor (#64)	N/D	48 Seg		
		gamera2-bpel	17 Hor (#85)	N/D	7 Min 17 Seg		
		gamera2-core	17 Hor (#97)	N/D	9 Min 36 Seg		
		gamera2-ggen	18 Hor (#64)	N/D	24 Seg		
		gamera2-gui	16 Hor (#109)	N/D	32 Min		
		gamera2-gui-dists	2 Mes 4 días (#26)	3 Mes 5 días (#3)	12 Min		
		mubpel	18 Hor (#106)	4 días 18 Hor (#92)	1 Hor 9 Min		
		ServiceAnalyzer	18 Hor (#67)	N/D	1 Min 17 Seg		
		sourcesfm-parent	18 Hor (#53)	N/D	43 Seg		
							

Hagamos una prueba

Demostración práctica

Contribuciones en congresos

- **Towards higher-order mutant generation for WS-BPEL**

Blanco-Muñoz, E., García-Domínguez, A., Domínguez-Jiménez, J.J., Medina-Bulo, I.
ICE-B 2011 - Proceedings of the International Conference on e-Business,
143-148, ISBN: 978-989842570-6

- **Propuesta de una arquitectura para la generación de mutantes de orden superior en WS-BPEL**

Emma Blanco Muñoz, Antonio García Domínguez, Juan José Domínguez Jiménez,
Inmaculada Medina Bulo
Actas de las XVI Jornadas de Ingeniería del Software y Bases de Datos,
537-542, ISBN: 978-84-9749-486-1



Conclusiones y trabajo futuro

Conclusiones

- La primera herramienta que genera mutantes de orden superior para composiciones WS-BPEL.
- Mejora GAMERA, que sólo trata con mutantes de primer orden.
- Fácilmente parametrizable y ampliable.
- Puerta abierta para generadores de mutantes de orden superior en otros lenguajes.

Conclusiones y trabajo futuro

Conclusiones

- La primera herramienta que genera mutantes de orden superior para composiciones WS-BPEL.
- Mejora GAMERA, que sólo trata con mutantes de primer orden.
- Fácilmente parametrizable y ampliable.
- Puerta abierta para generadores de mutantes de orden superior en otros lenguajes.

Trabajo futuro

- Integración con la interfaz gráfica de MuBPEL.
- Estudio de la calidad de los mutantes generados.
- Módulo de mutantes para Java.

Muchas gracias

¿Preguntas?

emma.blancomu@alum.uca.es